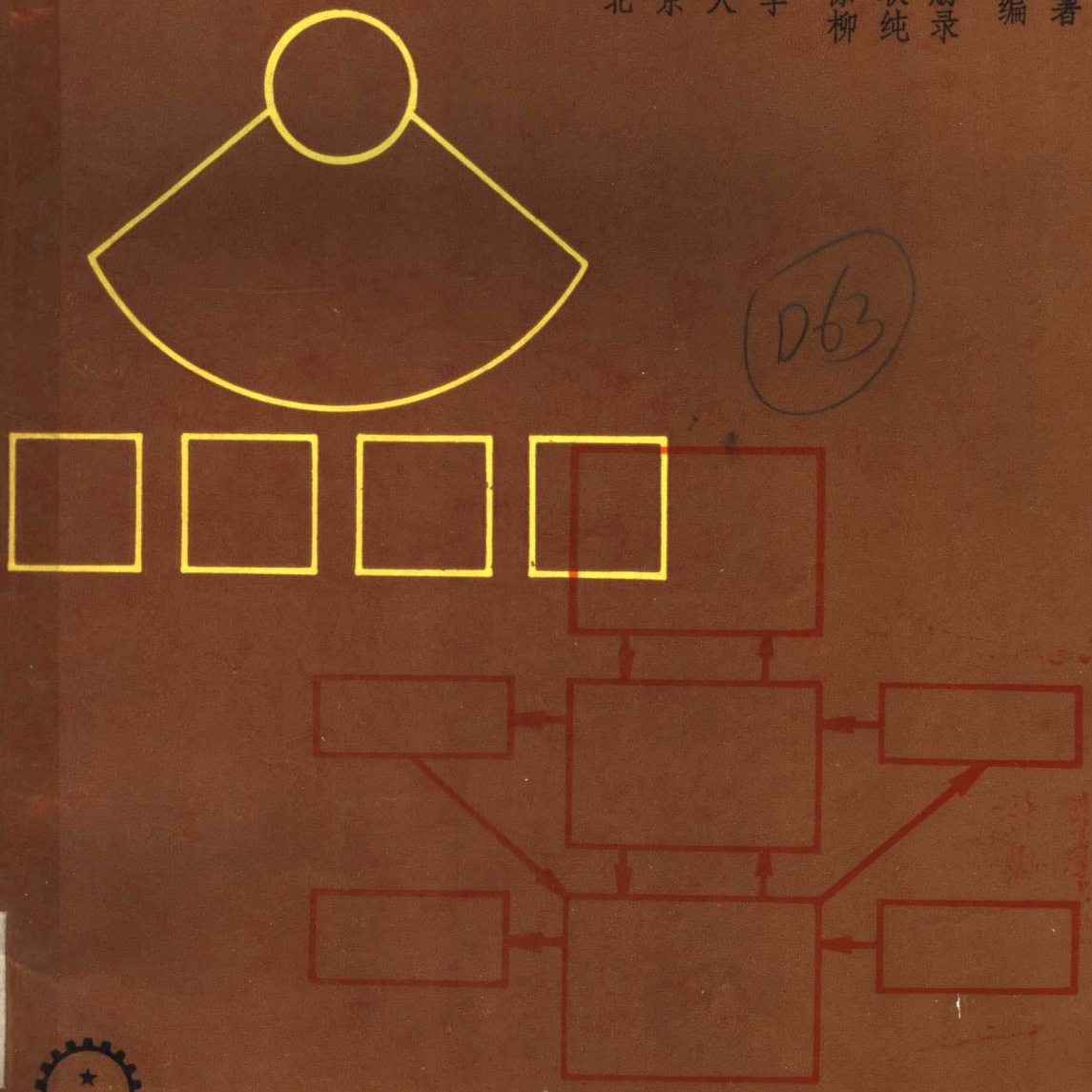
计算机操作系统原理

北京大学 徐联航 柳纯录 编著



机械工业出版社

计 算 机
操 作 系 统 原 理

北 京 大 学
徐联舫　柳纯录　编著

机 械 工 业 出 版 社

本书讲述操作系统基本概念和一般原理：第一章从内部和外部两方面对什么是操作系统作一般描述；第二章讨论操作系统核心；第三章是作业管理，第四章到第七章涉及空间资源的管理；第八、九章分别讨论系统结构和性能；第十章讨论操作系统的一个特殊问题——死锁问题；附录一是操作系统的硬件基础；附录二包含了几个流行操作系统的介绍。

本书适合于大专院校计算机有关专业作为教材，对从事计算机使用、研究、维护等工作的广大科技人员也有参考价值。

计 算 机
操 作 系 统 原 理

北 京 大 学

徐联舫　柳纯录　编著

•

机械工业出版社出版（北京阜成门外百万庄南里一号）
（北京市书刊出版业营业许可证出字第117号）
河北省三河县印刷厂印刷

•

开本787×1092 1／16 • 印张12 • 字数289千字
1985年12月北京第一版 • 1985年12月北京第一次印刷
印数 00,001—20,000 • 定价2.30元
统一书号：15033 • 6390H

前　言

操作系统已成为现代计算机系统最必要的环节之一，操作系统课程也已成为计算机有关专业的必修课程。总结操作系统一般原理，以指导当前广泛的实践活动，是很有必要的。本书总结了我们在这一领域从事生产、研究和教学的十多年经验。并且，书的基本内容曾以讲义形式在校内外经过一、二十遍教学实践，形成了一个比较稳定、成熟的教学体系。

本书主要介绍操作系统的基本概念和原理，而把有些内容留给高级课程或其它课程去处理（如数据库、微机、网络等）。另外，实际经验的获得应该通过实例或实习来解决，所以，书的基本内容不依赖于背景机。

全书内容分成十章：第一章从内部和外部概括介绍什么是操作系统；第二章介绍操作系统的核心，这是操作系统的关键部分。第三章是作业管理；第四章到第七章是有关空间资源的管理。第八、九章分别讨论系统的结构和性能。实际上，统一的结构观点和统一的性能观点是贯穿全书的，这两章是集中论述。第十章讨论操作系统的一个特殊问题——死锁问题。最后有两个附录。附录一是操作系统的硬件基础。计算机组织应当是操作系统的先行课程。这个附录把与操作系统有关的最必要内容作集中概括的介绍。读者最好先读这个附录再开始读正文。我们的教学活动基本上也是按这个次序进行的。附录二是几个流行操作系统的简介。它既是一般原理之后的实例，又是了解这些系统的入门向导。

上述内容中，第九章性能分析要求较强的数学基础，如果有困难，可以略去不讲。第十章死锁问题的理论模型部分也可不讲或少讲。这样、连同少数几次习题课和讨论，整个课堂教学大约是50学时，满足原教育部制订的操作系统教学大纲的规定。

实习是操作系统教学的重要环节。建议的实习题目是 A.C.Shaw [3] 书后的题目。一般要进行简化。

本书未附习题。我们已经积累了一些习题，正在整理，以后将有专门的一个习题集作为本书的补充。

本书是在反复研究和修改基础上形成的，并力图反映计算机操作系统方面的新发展、新成果，但书中仍可能存在不少缺点、错误之处，恳请广大读者提出宝贵意见。

本书写作过程中，我校杨芙清教授给予了热情指导和帮助，俞士汶同志、方裕同志及很多同学提出了建设性意见。校内外许多同志，为本书的出版作出了贡献。在此谨致衷心的感谢。

编　者

1985 年 6 月

目　录

第一章 操作系统概述

一、什么是操作系统

现代计算机系统由硬件和软件两部分构成。操作系统是系统软件的基本部分，它统一管理计算机的资源，协调系统各部分之间、系统与使用者之间以及使用者与使用者之间的关系，以利于发挥系统效率及方便使用。

操作系统是随着计算机体系结构和使用方式的发展而产生的。

早期的计算机在体系结构上是单控制中心，在使用方式上是单用户独占。所谓单控制中心是指中央处理机 CPU 是全机唯一的控制中心，无论内存，外存或外部设备都在它直接控制下工作。CPU的这种控制作用是通过执行 CPU 指令实现的。所谓单用户独占方式是一个用户使用计算机时，排斥其它用户，即他是计算机全部资源的唯一使用者。

分析表明，这样的使用方式对计算机资源的利用存在着惊人的浪费。

首先，单控制中心意味着CPU既管计算、也管传输。而由于 CPU 与外部设备的工作速度存在巨大差别，当CPU执行传输任务时，高速的CPU等待低速的外部设备，这时 CPU的利用率很低。例如，CPU 每秒执行一百万条指令，而打印机每秒打印10行，两者相差好几个数量级。计算机发明以后一段时间内，CPU 的速度有很大提高，而机械的外部设备其速度很难相应地提高，从而高速的CPU与低速的外部设备间的不匹配性越来越突出。

其次，单用户独占的使用方式也严重影响系统效率的发挥。由于从程序进入计算机到执行完毕下机，全部是手工控制，所以当人操作时，机器中的大部分资源，尤其是CPU，是空闲着的。另外，若程序执行中发生错误，需要人来思考判断查错修改，这时 CPU 也是空闲着的。由于人的操作与思考的低速度，造成机器资源利用上的浪费也很严重。

有人估计，在这样的系统中，CPU 的利用率只有7%。

现代计算机有两个特点：多部件并行与多任务共享。

我们用“联机多道程序系统”为例来说明。

首先，我们需要使用通道概念。通道是从 CPU 分出来专管输入输出的处理机。当需要传输时，CPU 只要驱动通道去完成就行了。由于通道只是用来控制传输，可以以较小成本来实现。另外，通过对通道结构格式的改进，还可以进一步使通道工作更有效①。

通道的特点是可以与CPU并行工作，即它有向内存独立存取数据的能力。当 CPU 驱动通道之后，通道就可以独立地与 CPU 并行工作。另外，多通道或多路通道意味着设备与设备可以并行工作。这样，当低速的外部设备工作时，CPU不必等待，从而发挥了 CPU 的效率。

但是，通道的设立只是提供了与 CPU 并行工作的可能性，而不是必然性。CPU与通道

---

① 参见附录一操作系统的硬件基础

能否并行工作，不仅取决于硬件的物理可能性，也取决于程序的逻辑可能性。对一道程序而言，有时不能做到这一点。例如，程序的输出信息往往是某种计算的结果，而输出信息缓冲区大小又很有限，可能在程序运行过程中要反复利用（如在循环段内打印）。这样，若上一次送到缓冲区的输出信息未输出完，则无法进行进一步的计算。多道程序的引入就是利用不同程序在计算与传输时间上的重叠而达到充分利用多部件并行的目的。

例如，假定系统有一个CPU，一台打印机。又系统中有两道程序，一道名为A，另一道名为B。若这两道程序对 CPU和打印机的利用是以图1.1所示方式进行的，那末CPU及打印机都能得到充分利用。当然，实际很少出现这样理想化的情况，但是，从统计上看，各道程序对并行部件的交叉利用情况是经常的。

图1.1中实线表示程序 A 对部件的利用，虚线表示程序 B对部件的利用。

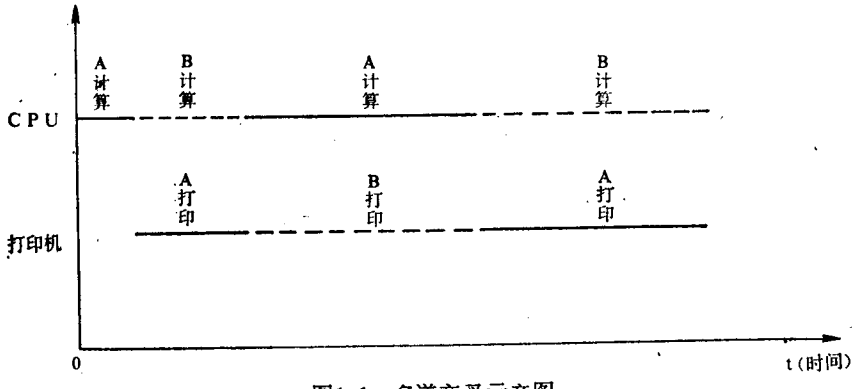


图1.1 多道交叉示意图

上面讨论了计算与传输的并行性，下面讨论计算与控制的并行性。

所谓“联机”多道，是指用户可以利用联机终端对自己的程序进行现场控制。这里也同样利用多道交替来避免一道程序控制“间隙”中CPU的闲置。终端也通过通道与CPU连接，所以终端与 CPU 可以并行工作，各终端之间可以并行工作。见图1.2。

需要指出，中断系统的引入，实现了 CPU 与通道的联系，实现了人与计算机的通信。因而，它也是现代计算机必不可少的硬件设施。

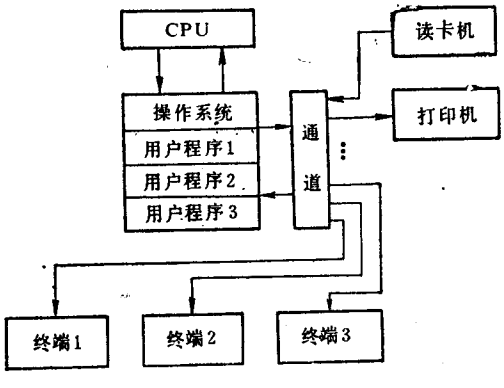


图1.2 联机多道程序系统

为了提高系统效率，还可以有其它进一步措施。不过上述系统已经有了现代计算机的基本特征，即多部件并行与多任务共享。多部件并行是提高效率的基础，如果没有这一点，多任务共享并不能提高效率。例如，如果只有单控制中心 CPU，假定有两个程序，分别要求 CPU 时间（包括计算和传输时间）为 3 分钟和 5 分钟，那末，不论怎样安排，总共需要 8 分钟时间。安排的不同只可能使两个程序完成早晚有不同。多任务共享是对多部件并行的利用。

多道程序的实现需要解决一系列问题。一般说来，对多任务系统要解决：怎样实现多任务共享？如何使这种共享能充分利用系统资源？如何防止各共享任务之间的冲突？

由于各用户之间是相互独立平等的关系，一个用户没有权利和义务来管另一个用户。所以，正像任何机关或企业需要有行政管理机构一样，计算机中也要有公共管理机构来处理共享。这是通过程序来实现的，这个程序的任务就是管理计算机系统各种资源，控制、协调各用户对这些资源的利用，并使之有效、合理，这就是操作系统。操作系统还具有让用户更方便地使用计算机的功能。例如，外部设备的使用要经过复杂的驱动过程，用户很难直接使用。经过操作系统，使用就大为简化。

所以，操作系统是计算机系统的中心控制软件，它的职能是组织多个用户对计算机各种资源的共享，并使之有效、合理和方便。

所谓合理性，包含两方面意思：

1.响应时间

即考虑到各用户程序到达系统的先后次序，也就是考虑先来后到。操作系统不保证先进先出，但也不应过分悬殊。特别要防止个别用户程序被无限期推迟的情况。

对分时系统，“及时响应”有另外含义，我们将在以后讨论。

2.保　护

这里指用户程序之间的相互隔离，防止一个用户程序的错误影响其它用户程序。

总之，由于合理性和方便性，操作系统向每个用户提供的是一台“虚机器”。也即每个用户好象自己有一台单独的机器，感觉不到其它用户的存在；而且这台虚机器比真实机器使用更方便。也可以说，操作系统实现了将一个物理机器转化为多个逻辑机器。一般虚机器比真实机器速度慢和容量小。

我们把建立在通道和中断系统基础上的多道程序系统作为讨论现代操作系统的出发点。

二、操作系统的特点

操作系统是比较复杂的系统软件，不仅设计一个操作系统绝非容易，甚至理解一个操作系统也往往有很大困难。因此，先了解一下操作系统的特点是有好处的。

操作系统的特点主要有如下几点：

1.并　行

如果我们把硬件与软件统一考虑，那末这种并行性表现为：

1)硬件与硬件并行。如CPU与通道并行，子通道与子通道并行；

2)作业的计算与传输并行；

3)作业与作业并行。如多道程序；

4)系统程序与系统程序并行。

其中，1)是基础，2)、3)建立在这个基础上。4)又是为2)、3)服务，它的并行性是2)、3)并行性的反映。

并行性是操作系统的主要特征。

2.共　享

指的是多个任务对计算机资源的共享。这种共享是并行共享。困难仍然在于并行性。

共享的目的是为了提高计算机资源的利用率。操作系统的一个重要职能就是组织好对资源的共享，以提高系统效率。

要实现资源共享，需要解决一系列问题。如资源分配或资源的“互斥”利用、公用资源以及对任一共享者的保护，等等。

3.随机性

操作系统是含有随机性的系统程序，在这一点上，不仅与一般的用户程序不同，而且与编译程序等其它系统程序也不同。

中断发生时间的随机性就是操作系统随机性的重要实例。操作系统必须随时（可能有延迟，但不改变随机性）响应中断，而且要保证整个功能的正确性。所谓操作系统的非确定性，就是指的这个意思。即它不是事先规定各种事件何时发生，而是事先安排好对各种可能事件的处理——不管这些事件何时、以何种次序及以何种组合方式发生。

随机性的其它例子是作业到达和离开系统时刻的随机性、作业对资源要求的随机性等。例如，若允许作业在执行过程中根据需要随时提出对资源的申请，则这种随机性给操作系统带来资源管理上的很大困难——一方面只有系统随时都拥有充足的空闲资源才能保证对用户随机申请的及时满足；另一方面，要达到系统的高效率，就必须使系统所拥有的资源总是处于充分忙碌的状态。调节这两者关系是操作系统基本问题之一。

随机性所以造成复杂性，主要由于这种随机性是一种与并行以及共享有关的随机性。并行活动以随机方式相互影响——相互联系及相互制约。在顺序系统中也可以说有随机性，如一个程序中各指令执行速度是随机的，但这不影响整个程序的逻辑，因为我们只关心每一条指令的功能以及各指令顺序执行所实现的功能。而在并行系统中，各顺序指令流之间的相互关系就复杂得多。

掌握多个并行活动比掌握单一的顺序活动复杂，这是符合人们常识的。但是，并行活动仍然是可以掌握的。操作系统中的并行性也是可以掌握的，问题是要有正确的认识，并采用恰当的方法。

三、操作系统的环境

以上我们讨论了为什么引入操作系统与什么是操作系统。这里我们通过考察操作系统与外部环境的关系来了解操作系统。

操作系统的外部环境主要是硬件、其它软件以及“人”。

1.操作系统与硬件的关系

一般来说，软件是建立于硬件基础上，是对硬件功能的延伸。操作系统与硬件关系尤其密切。为了实现操作系统，必须有相应的硬件设施。但是，这些界面没有统一、整齐的形式。对编译程序，指令系统就可以表示它所工作的硬件基础。而对操作系统，除了指令系统，还有通道与中断，以及其它的硬件设施(以分散形式与操作系统相应部分衔接)。而且，操作系统与相应硬件联系是很密切的。例如，中断系统就是由硬件与软件两部分配合共同完成的。

一般说来，软硬件界线的划分是有灵活性的，但这种灵活性是相对的。有些工作必须由

硬件来完成，软件无法代替，如交换程序状态字。

操作系统的功能必须与硬件基础相匹配。当硬件条件不具备时，追求过高功能，会使操作系统本身开销太大，收效太小或得不偿失（操作系统的目的是为了提高计算机资源利用率，但操作系统本身也需要计算机资源，包括CPU时间及内存空间等，这叫“开销”）。例如，一个计算机有内存空间M，如果操作系统本身占了M/2，那末，即使这个操作系统充分起作用，充其量也只是使另外M/2的内存空间利用率达到100%，整个内存空间利用率只是50%。操作系统功能太低也不利于系统效率的充分发挥。

2.操作系统与其它软件的关系

操作系统是整个计算机系统的控制管理中心，其中也包括对其它各种软件的控制和管理。例如，连接装配程序、编辑程序、编译程序，运行程序和各种软件工具等。这时，我们可以把这些软件看作操作系统的子系统，由操作系统统一控制和调度。在此基础上进一步开发应用软件和应用程序。操作系统的控制作用使它比其它软件有更高的“地位”，这种地位一般需要通过一定硬件设施来保证。

图1.3表示了从硬件到应用程序的结构层次。

（图中系统软件指除操作系统外的其它系统软件）

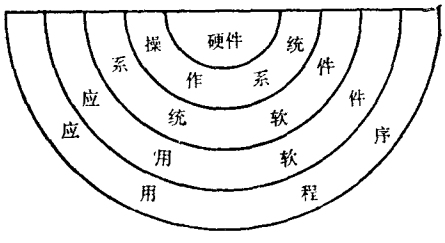


图1.3 计算机系统结构层次图

3.操作系统与“人”的关系

这里的“人”指程序员、操作员和管理员。

操作系统与“人”的界面中，主要是与程序员的界面。

程序员可以理解为应用程序的编制者。操作系统与程序员的界面就是操作系统向程序员提供的功能。程序员主要关心的是使用系统的方便性与合理性。系统提供的方便性越多，程序员编制应用程序就越容易。操作系统和其它软件提供了程序员设计、调试、运行和维护应用程序的方便性。

操作系统起着计算机系统控制者的作用，它实现了计算机自身管理的自动化，但是人仍然是最终控制者。这里所说的“人”就是指的操作员。操作员是：

1）系统初启者；

2）（管理）政策实施者；

3）（系统）状态监督者。

4）运行维护者；

5）使用服务者。

操作系统提供的自动化程度越高，操作员的工作越简单。操作系统与操作员的界面也是有灵活性的。有的事情可以让操作系统做，也可以让操作员做，这可根据必要性与可能性，机器和人的特点等权衡决定。

计算机装置的管理者有以下任务：

1）制定系统政策。如决定让哪类作业优先。

2）制定收费标准。如对不同的资源要求及服务质量收费不同。

3)系统的维护、监督和改进。如为了改进系统性能而搜集统计数据。

操作系统为管理提供的功能反映在操作系统算法中，或者通过专门程序来实现。

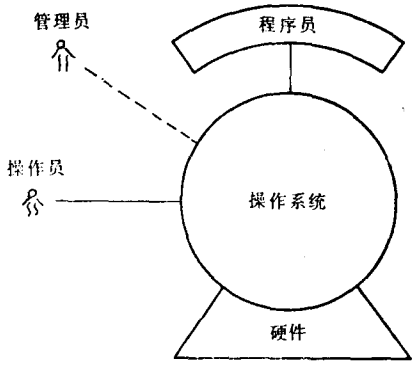


图1.4 操作系统的环境

最后，我们用图1.4来表示操作系统与环境的各个界面，这些界面相互联系又相互制约。一件事，由硬件实现，还是由操作系统实现；让操作员做，还是让程序员做；这是有灵活性的。操作系统的设计者必须根据具体情况调整这些界面。也就是要权衡系统的有效性与使用的合理性等因素。

图1.4中，我们只标志操作系统。不妨把它看作广义的操作系统，即包括通常意义的操作系统、系统软件和应用软件。但操作员主要与“狭义”操作系统打交道。

四、几种典型的操作系统

以上对操作系统从内部原理及外部关系进行了初步讨论。为了对操作系统功能和原理有进一步了解，下面介绍几种典型的操作系统，分别说明它们的使用特点及基本原理。

1.批量系统

又称批量处理系统、批处理系统，也叫作业流处理系统。本书中除个别情况外，一般都统一使用“批量系统”一词。

批量系统一般与spooling技术相联系，所以也叫spooling系统。

spooling技术的基本原理是利用外存作为输入输出缓冲，使输入、计算和输出并行。另外，作业进入系统也通过外存缓冲，即作业先从输入设备到外存，再从外存到内存。用于输入输出缓冲的外存称为“井”。用于输入缓冲的外存称为输入井，用于输出缓冲的外存称为输出井。输入输出井的作用是调节计算与传输的不协调性。没有井，一道作业的计算与传输可能彼此等待，造成CPU或外设的闲置。有了井，就可以缓和这种状况所造成的浪费。另外，输入井对作业的缓冲也有利于减少各用户上下机交接时间。一般从一个作业结束到下一个作业进入系统要经过一个输入操作过程，中间有时还可能因发生故障而重新开始，在这段时间内机器资源未充分利用。作业缓冲使作业B的输入过程与作业A的运行过程并行起来，即使得输入井中保持一定数目的缓冲作业，当需要调入新作业时从输入井中选取，不必现等从输入设备输入。这时目的不是调节一道作业计算与传输的不协调性，而且调节作业到达的不均匀性，或者说是调节作业流的不均匀性。作业缓冲还提供了调度的灵活性，即作业执行的次序不一定按照进入系统的先后，而是可以从作业缓冲中选取，以便获得更好的系统性能。

批量系统的使用特点是作业从输入设备进入系统后直到结束（正常结束或异常结束），不允许用户对自己作业的控制和干预。这有利于提高系统效率，但不便于程序的调试。

spooling系统中，利用外存作为输入输出缓冲的方法也叫脱机输入输出，即入/出设备不再与用户程序直接联系。不过，如图1.5所示，这里采用所谓“假脱机”方法，即入/出设备与

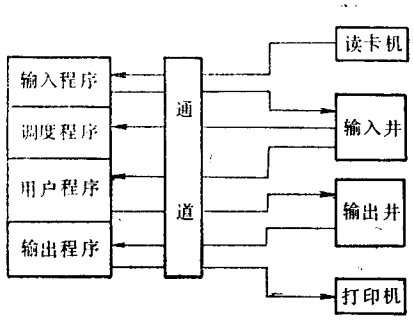


图1.5 spooling系统流程图

井的联系仍然通过同一台计算机实现，只是逻辑上输入程序、调度程序、用户程序和输出程序是相互独立的。真脱机是由单独的计算机负责作为井的外存与入／出设备之间的数据传输。

2.分时系统

分时系统中有多个终端，用户通过终端与系统交往。一般，一个终端是一个带有屏幕显示的键控设备。

批量系统虽然提高了系统效率，但由于限制了用户与自己程序的交往，所以不利于用户对程序的调试。分时系统也是建立在多任务共享与多部件并行的基础上，但它为用户提供的使用方式是完全不同的。分时系统中，允许多个用户同时通过各个终端与系统直接交往，随时修改、调试自己的程序。

分时系统有三个特点：

1)交互性

在分时系统中，程序员一般通过在终端上输入命令工作，要求系统对输入的命令及时响应。所以分时系统也叫分时会话系统，即它便于人——机会话。什么叫“及时”？这是以人的心理感觉为尺度的。例如，2～5秒钟一般认为是合理的响应时间范围。

2)多路性

分时系统一般带有多个终端，从而可以同时允许多个分时用户工作。大型分时系统可以带有几百个终端，这是因为程序员思考，操作的速度很慢。分时用户过少，CPU空闲机会增多；分时用户过多，又会使响应状况变坏。

3)独占性

前面已说过，操作系统实现了向各用户提供一台虚机器，并且各用户的虚机器互不干扰。在分时系统中，由于每一用户都可以通过一台终端设备与系统直接联系，加之响应及时，使用户感到好象系统只为他一个人服务一样。

上述三点中，最主要的是交互性。为保证及时响应，采取轮转算法分配CPU，即以循环方式依次为每个用户的请求分配一个“时间片”，这也是名称“分时”的由来。这样做的作用是防止为一个用户的服务时间过长垄断了CPU而妨碍对其它用户请求的响应。

响应时间与多种因素有关：CPU的速度、用户请求的种类、系统连接方式或正在工作终端的数目以及调度算法（如时间片值的选取）等。

下面谈谈时间片大小的选取。时间片过大，可能会由于要求时间长的服务请求挡住了队列，增大平均响应时间；时间片过小可能会增加各服务请求循环排队次数，也增大平均响应时间，而且系统开销也增大。例如，有三个请求A、B、C，按先后次序要求CPU时间单位为10，2，2。我们考虑取三种不同时间片，看每种的平均响应时间：

1)时间片取10，平均响应时间为（10＋12＋14）/3＝12

2)时间片取2，平均响应时间为（4＋6＋14）/3＝8

3)时间片取1，平均响应时间为（5＋6＋14）/3＝8.33

计算过程如图1.6所示

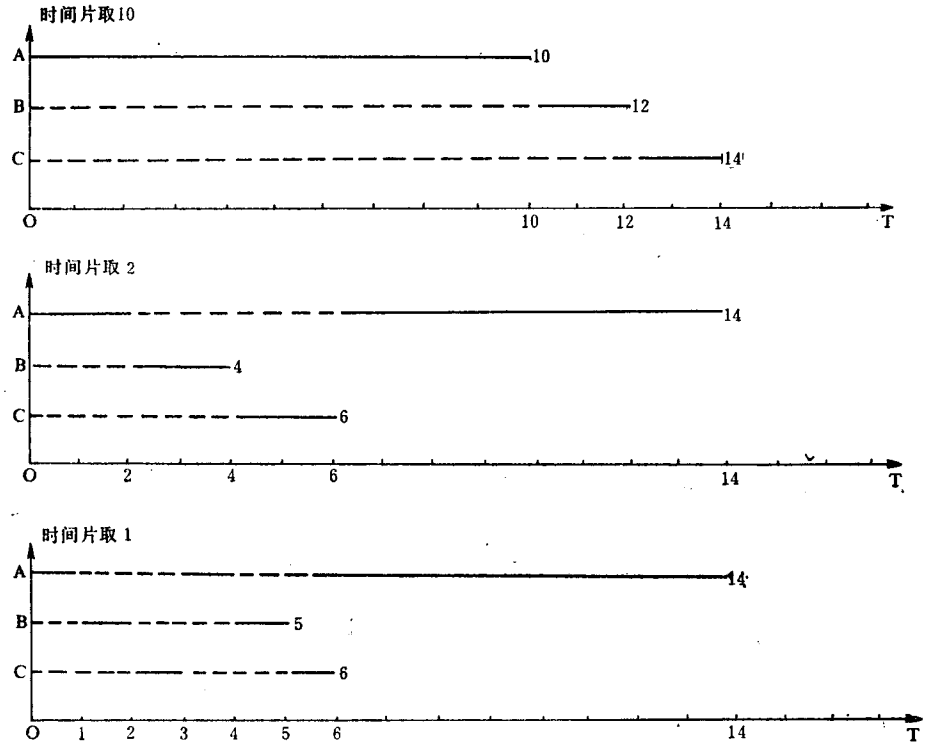


图1.6　响应时间对照图

这里时间片为2时，平均响应时间比时间片为10或为 1 时短。

批量与分时代表计算机使用方式的两个极端，即人的干预程度最少与最多。联机多道介于两者之间。联机多道并不强调及时响应，另外，交互“深度”也不如分时系统。联机多道允许程序员与操作系统会话，而分时系统还允许程序员与程序语言会话。为此，分时系统的程序语言应当是具有会话功能的语言。如Basic语言，“会话FORTRAN”语言。它们一般是解释执行的。

终端设备，轮转算法和会话语言是实现分时系统的必要条件。由于轮转算法以及会话语言的解释执行，系统开销较大。所以分时系统一般适合于小作业或程序的调试。

为了提高分时系统的效率，还可以采用对换技术，也叫滚入滚出。这是着眼于提高内存利用率。当某个分时作业暂不运行时，把它调到外存，然后在某个适当的时刻再调回内存，继续运行。

例如，有一分时系统支持25个终端，内存只放一个分时作业，其余都在外存。每当一个作业接受一个时间片服务后，就将它送回外存，再从外存调入下一个作

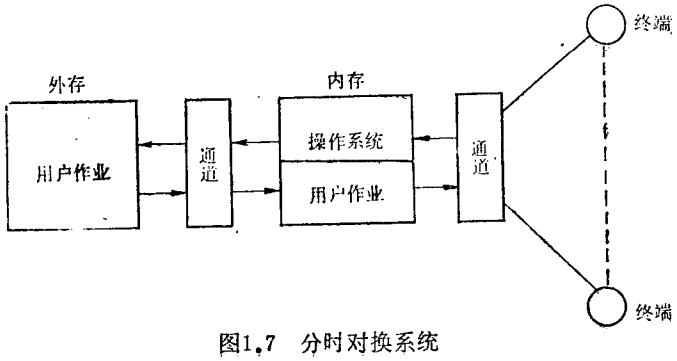


图1.7　分时对换系统

业。如果时间片为40ms，作一次内外存对换时间也是40ms。易知对每个小于一个时间片的请求，将在 25 × (40ms + 40ms) = 2 秒之内获得响应。

上述系统中，每当作业对换时，CPU闲置，所以CPU利用率不超过50%。为此，可以在内存放多个分时作业。这样，当一个内存作业参加对换时，可以执行另外的内存作业。

为了避免单纯的分时系统造成系统效率的损失，还可以采取分时与批量结合的方法，即系统既支持分时作业，也支持批量作业。这时分时作业称为"前台"作业，批量作业称为"后台"作业。CPU 可以采取轮转与优先结合的算法。例如，对分时作业和批量作业的关系，分时作业优先，尽量先运行分时作业。当分时作业无请求时才运行批量作业。一般计算中心往往白天有很多分时用户，晚上就很少。所以批量作业在晚上获得更多的运行机会。在各个分时作业之间，则可以轮转分配CPU。

批量系统与分时系统的共同特点是"作业处理"，也就是用户以作业为单位使用计算机。作业有开始和结束，不同作业之间相互独立。另外，批量系统与分时系统往往还都建立于多道程序基础上，即批量多道与分时多道。下面要介绍的实时系统与作业处理系统是不同的。本书主要讨论作业处理系统。

3.实时系统

实时系统包括实时过程控制和实时信息处理。

实时系统与作业处理系统不同。在作业处理系统中，操作系统为各个用户作业服务，它本身没有要完成的"任务"，只是起着控制、服务的功能，所以是"通用系统"。而实时系统一般是"专用系统"，即实时操作系统中包含着特定的应用程序，统一实现特定的控制、服务功能。所以，它不包含"作业"或"道"的概念，而只有固定的若干"任务"程序，它们配合起来，反复处理来自现场的数据或信息。换句话说，作业处理系统的处理对象是作业，而实时系统的处理对象是数据或信息。

实时系统要及时接收来自现场的数据，及时加以分析处理，以及及时作出必要的反应。

实时控制系统还可以分为两类：一类是以计算机为控制中枢的生产自动化系统。如冶金、发电、化工、炼油的自动控制。其中往往要有专门的模数及数模转换设备。模数转换设备把生产过程中采集的物理量转化为数字量，送给计算机。而数模转换设备则是把计算机发出的数字量转化为物理量，回到生产过程。另一类是飞行物体的控制，如飞机、导弹、人造卫星的制导。这时计算机可以安装在飞行物体上，也可以在地面遥控。

实时信息系统利用计算机并通过众多的联机通信线路完成对信息的接收、存储、检索、修改、加工、删除、传送等功能。

实时信息系统往往与数据库系统相联系。

实时信息系统的联机通信线路可以是有线的，也可以是无线的。联机通信线路的终点可以是实时终端，也就是人，也可以是其它计算机。一般地，实时信息系统可以是由若干计算机与若干终端通过通信线路连接起来的网。

实时信息系统的例子是情报检索、飞机订票、银行业务、电报电话交换等。

图1.8和图1.9分别是这两类实时系统的模型。

实时系统的特点是：整体性、及时性与安全可靠性。

整体性是指实时操作系统中各程序互相配合完成共同的任务。例如，在飞机订票系统

中，当有一个订票请求时，系统中有的程序对请求合法性进行检查，有的程序分析请求内容，有的程序驱动外存等。整体性的另外含意是系统维持着一个公用数据库。对这个数据库的任何一个增删修改必须为这个数据库的其它访问者所知。例如，在飞机订票系统中，当一个订票请求被满足时，必须对反映销售情况的数据库进行更新，以免对同一张票订重。

及时性是由于信息来自现场，往往不可重复，如不及时接收，会造成信息丢失。

安全可靠性是由于实时系统是在现场进行直接控制、处理，一旦发生错误往往会造成不可挽回的后果及重大损失。

实时信息系统的终端用户可以是普通用户，他们不必是程序员，而只要掌握有限的操作命令即可。如各种信息询问系统终端。

实时系统的操作员也就是系统的监控员。他们可以通过专门的控制终端及控制命令与系统通信。

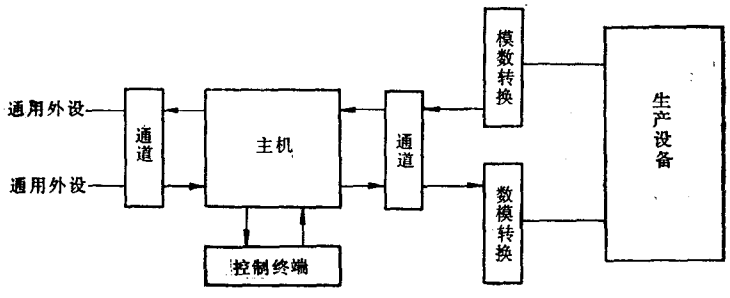


图1.8 实时控制系统

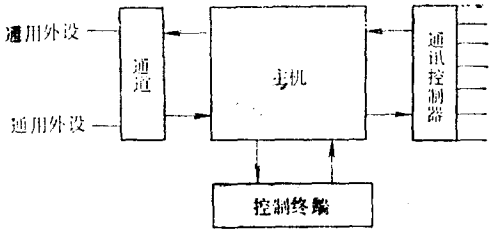


图1.9 实时信息系统

第二章 系统核心

前面对什么是操作系统作了一般介绍，其中讲到并行性是操作系统的主要特点，本章将对并行性进行深入的讨论。

并行任务之间会发生种种关系。并行程序共享CPU资源，表现为进程调度；并行程序的竞争与合作关系，表现为进程通讯。进程调度，进程通讯与中断处理、时钟管理等，构成了操作系统的核心。

核心是操作系统的基本成分。各个操作系统功能各异，结构也多不相同，但核心却是任何操作系统必不可少的成分。尽管目前还没有通用、标准的操作系统核心，但各个具体的操作系统一般都把它的基本部分叫做核心。所以，我们这里主要强调的是核心的基本概念，与任何一个具体的操作系统核心不一定一致。

我们从多道程序出发来说明并行程序。

首先，并行是指各用户程序之间的关系，即多道程序可以看作多个并行程序。其次，当一道用户程序执行过程中要求传输时，通过“启动入出”指令驱动通道（或包括相应子通道，下同），从而通道与CPU并行工作。但事实上，由于驱动通道是个复杂过程，不是一条启动入出指令就够了。为了方便用户，节省内存（也为了防止各用户程序同时驱动通道而发生冲突），驱动通道的工作是由系统中专门程序来完成的，我们不妨把它称为“启动入出程序”。用户程序要求传输时，只要提供有关参数，如要传输信息的位置，数量等，然后“驱动”启动入出程序即可。启动入出程序的功能是按照用户程序要求组织通道指令链，执行启动入出指令等。通道执行传输任务可能要较长时间，用户程序不必等待，可以继续在CPU上执行，传输结束后的处理也由启动入出程序负责，最后才“报告”用户程序。也就是用户程序一当驱动了启动入出程序就可以继续执行。这时用户程序可以与启动入出程序并行。这是CPU可以与通道并行的反映。

一个操作系统中的并行程序之间往往会有种种联系（其中有些是直接的联系，有些是间接的联系，我们只考虑直接的联系）。例如，几个用户程序对某个公共资源的竞争，或者用户程序与启动入出程序的合作。由于各并行程序是相互平等的，需要有公共机构来处理这种联系，以协调它们的活动，这就是并行程序的通讯问题。

任何一个程序的运行需要有一定的资源。首先是CPU和内存资源，还可能要求其它资源。其中如内存资源的获得可通过固定分配方法或通过一个内存管理程序获得（内存管理程序本身所需的内存用固定分配方法获得）。但CPU资源只能通过动态方法获得，即通过一个调度程序来分配。调度的必要性也是由于系统中存在着并行程序而引起的。所以，调度即指并行程序的调度。

本章我们先讨论并行程序的特性，然后再讨论并行程序的通讯与调度，最后讨论系统核心中的若干问题。

一、并行程序——进程

大型软件的出现及日益复杂化，使程序设计的概念和方法有了很大的发展。例如，从操作系统的实践和研究中引出的“并行程序”概念便是其中之一。为了讨论并行程序，我们先来讨论顺序程序，以便比较。

1.顺序程序

顺序程序就是程序，因为顺序性是我们一开始接触“程序”这个概念时自然包含的性质。我们把一个“程序”理解为“一个在时间上按严格次序前后相继的操作序列”。

这里的“操作”可以是高级程序语言中的语句，或机器语言中的指令。

程序的顺序性与计算机硬件的顺序性是一致的。因为CPU是通过时序脉冲的控制顺序执行指令的。而我们把要计算的问题表示为程序也正是为了便于CPU的执行。所以这时程序既是对计算过程的描述，也是对CPU动作过程的描述（对用高级程序语言写的程序，不是对CPU动作过程的直接描述，但语句与指令有着顺序性的对应关系，即一个语句对应一串指令）。

在顺序系统中，每一时刻最多执行一个操作。如果我们把一个程序的执行过程看作一系列状态转移过程，也就是每执行一个操作使系统从一个状态转移到另一个状态。那末，顺序系统中状态是时间的单值函数（这里时间是离散变量）。

比如，如果有一条指令

K； add A

意味着将地址A的内容与累加器L相加，并将结果保留在L。执行这条指令后，系统状态改变如下：

L = (L) + (A)

PC = (PC) + 1

这里PC为程序计数器（或叫指令地址计数器）。

当问题涉及到一个大程序时，可以引进子程序概念。子程序有两个作用：结构清晰和节省存储。

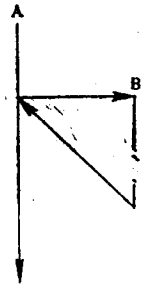


图2.1 子程序

例如，图2.1中程序B是程序A的子程序，程序A则是程序B的主程序。

子程序概念有相对性。例如，图2.2中程序B是程序A的子程序，但程序B是程序C的主程序。即子程序可以嵌套。

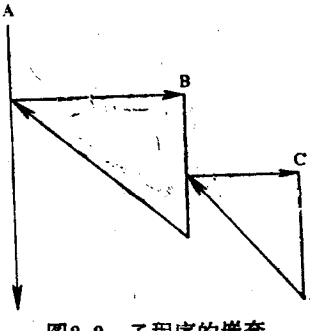


图2.2 子程序的嵌套

主程序与子程序虽然相互依赖而存在，但并非必定一一对应。一个主程序可以有多个子程序，一个子程序可以有多个主程序。例如，

图2.3中，程序B_1，B_2分别是程序A的子程序；程序C′既是程序B′的子程序，又是程序A′的子程序。

对于顺序程序，无论怎样复杂的结构，都可以化为逻辑上等价的单一程序。例如，图2.3可以化为图2.4的形式。